

Esercitazione Java e ADT

Recupero di informatica - 4αII, $\alpha \in \{A, B, C\}$

05 luglio 2022

Si risolva il problema proposto chiarendo, all'occorrenza, alcuni dettagli risolutivi non facilmente intuibili.

Problema

La società “*Curre curre guagliò*”, società Veneta di fitness, organizza una maratona amatoriale nel comune di Valdobbiadene. Per ogni categoria mette in palio 3 premi. Rispettivamente i premi sono:

- i al primo va un magnum di prosecco e una sopressa col filetto;
- ii al secondo va un processo sui lieviti;
- iii al terzo va un salamino nostrano;
- iv a tutti gli altri, come premio di consolazione, vanno degli hamburger vegani.

Richieste

Il software deve essere in grado di:

1. gestire una coda per l'iscrizione degli amatori;
2. registrare gli amatori in coda ed inserirli come gareggianti (sarà richiesto nome, cognome, età, distanza da percorrere tra $0,5km$, $7,5km$, $15km$: il numero identificativo dell'atleta sarà assegnato dalla sua posizione in coda);
3. registrare il tempo di un atleta dopo averlo trovato attraverso il suo n.ro identificativo;
4. visualizzare gli iscritti in una categoria;
5. stilare la classifica per ogni categoria di maratona una volta che **TUTTI** gli iscritti hanno terminato la prova.

Si suggerisce di definire una coda a lunghezza finita con un gestore definito da voi in un array statico, dove ogni nodo ha come unico attributo un numero intero: la tab che "gestisce"¹ la coda semplicemente inserisce in coda il primo numero successivo. Al momento dell'iscrizione, si va a scaricare la coda (e conseguentemente diminuirne la lunghezza).

Si suggerisce di scaricare la coda con un tasto che inserisce da solo il campo nella tab di registrazione (campo imm modificabile). Se questo campo non è vuoto, non si compie nessuna azione; se la coda è vuota, si mandi a video un messaggio di errore. Una volta salvato, questo sarà inserito

¹ricordatevi le primitive: *inserisci*, *elimina*, *lunghezza*.

in un *ArrayList* o in una *LinkedList*.

Per la visualizzazione degli iscritti in una categoria, basta inserire le voci in una *ListView* o in un *JTable*. In alternativa, populate una *JTextArea*.

Per la classifica, basta inserire le voci in una *ListView* o in un *JTable*. In alternativa, populate una *JTextArea*.

Passi per la risoluzione

Si produca:

1. l'albero gerarchico delle classi definendo le varie classi in UML, dove per ogni elemento sono esplicitati attributi, relazioni e tipi, seguendo i principi di programmazione orientata agli oggetti e di ADT. In caso di ereditarietà², fornire le definizioni di classe astratta, super classe, classe ereditata, overloading e overriding. Non è richiesta la definizione dell'albero gerarchico della coda;
2. dato l'albero del punto sopra, fornire l'implementazione delle classi precedentemente descritta in *Java*;
3. fatto il punto precedente, si considerino le richieste del precedente paragrafo: di questi si progetti un'interfaccia grafica (anche usando le tabbed pane) per soddisfare le richieste in *MVC*;
4. (★) Si salvino i vari record prodotti su un file.

Lasciate il punto contrassegnato con “★” come **ultimo ed opzionale**.

Si ricorda che la seguente esercitazione, seppur più complicata di quanto potrebbe capitare in prova di recupero, ricalca quello che può essere richiesto.

²l'utilizzo dell'ADT è fortemente consigliato, viste le ultime nostre lezioni...